

Основы программирования

ФИСТ 1 курс

Власенко

Олег

Федосович

Лекция 7.2

Массивы в Си (Продолжение)

Поиск максимального элемента

```
// поиск max
int max = a[0];
i = 1;
while (i < 5) {
    if (a[i] > max) {
        max = a[i];
    }
    i++;
}
// печать max
printf("max = %d ", max);
```

Поиск максимального элемента

Трассировка

```
// поиск max
int max = a[0];
i = 1;
while (i < 5) {
    if (a[i] > max) {
        max = a[i];
    }
    i++;
}
// печать max
printf("max = %d ", max);
```

Вход: $a[] = \{4, 3, 6, 2, 5\};$

Максимальный элемент поменять с начальным элементом

Вход: $a[] = \{4, 3, \underline{6}, 2, 5\};$

Выход: $a[] = \{\underline{6}, 3, 4, 2, 5\};$

Максимальный элемент поменять с начальным элементом

```
// поиск max
int max = a[0];
int iMax = 0;
i = 1;
while (i < 5) {
    ...
    i++;
}

{
    int tmp = a[0];
    a[0] = a[iMax];
    a[iMax] = tmp;
}
```

Максимальный элемент поменять с начальным элементом

```
// поиск max
int max = a[0];
int iMax = 0;
i = 1;
while (i < 5) {
    if (a[i] > max) {
        max = a[i];
        iMax = i;
    }
    i++;
}
{
    int tmp = a[0];
    a[0] = a[iMax];
    a[iMax] = tmp;
}
```

Сортировка массива

```
int a[] = {7, 3, 6, 2, 9, 1, 4, 8};
```

```
int a[] = {1, 2, 3, 4, 6, 7, 8, 9};
```

```
int a[] = {9, 8, 7, 6, 4, 3, 2, 1};
```

Чем отсортированная информация лучше неотсортированной?

Сортировка массива

Сортировка выбором

https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BE%D0%BC

Шаги алгоритма:

- находим номер минимального значения в текущем списке
- производим обмен этого значения со значением первой неотсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
- теперь сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы

Задача: Реализовать сортировку выбором

Вход: **(по возрастанию)**
`int a[] = {7, 2, 3, 6};`

Выход: `int a[] = {2, 3, 6, 7};`

Сортировка выбором

Блоксхема. Трассировка

Вход: `int a[] = {7, 2, 3, 6};`

Выход: `int a[] = {2, 3, 6, 7};`

Сортировка массива

Сортировка пузырьком

https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC

Шаги алгоритма:

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются $N - 1$ раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде. Отсюда и название алгоритма).

Задача: Реализовать сортировку пузырьком

Вход: **(по возрастанию)**
`int a[] = {9, 8, 7, 1};`

Выход: `int a[] = {1, 7, 8, 9};`

Сортировка пузырьком Блоксхема. Трассировка

Вход: `int a[] = {9, 8, 7, 1};`

Выход: `int a[] = {1, 7, 8, 9};`

Домашнее задание

1. Заставьте работать весь код, который написали на лекции.

2. Делайте домашние задания по лабораторным работам!